Claims

What is claimed is:

1. A method for analyzing behavior of a software system, comprising:

collecting details associated with a program task associated with said software system; and

providing said collected details for analysis.

2. The method of claim 1, wherein a duration of said program task is defined by one or more conditions associated with a state of said software system.

3. The method of claim 2, wherein said one or more conditions includes an entry or exit of at least one specified method.

4. The method of claim 2, wherein said one or more conditions includes a creation or deletion of at least one specified object.

5. The method of claim 2, wherein said one or more conditions includes an invocation of at least one specified object.

6. The method of claim 2, wherein said one or more conditions includes a passing of at least one specified object or scalar value as an argument, return value or field value.

7. The method of claim 2, wherein said one or more conditions includes at least one specified sequence of method invocations.

8. The method of claim 2, wherein said one or more conditions includes at least one specified resource exceeding at least one specified threshold.

9. The method of claim 1, wherein said collected details include an existence or sequence of specified method invocations.

10. The method of claim 1, wherein said collected details include an existence or sequence of specified object creations and deletions.

11. The method of claim 1, wherein said collected details include an existence or sequence of specified class loading and unloading.

12. The method of claim 1, wherein said collected details include values of specified arguments to invocations of specified methods.

13. The method of claim 1, wherein said collected details include values of specified return values from invocations of specified methods.

14. The method of claim 1, wherein said collected details include values of specified field values for invoked objects or field values for passed arguments.

15. The method of claim 1, further comprising the step of collecting said details for at least one specified number of task instances.

16. The method of claim 1, further comprising the step of collecting said details for at least one specified number of threads.

17. The method of claim 1, further comprising the step of dynamically modifying said program task specification associated with said analysis in an iterative process.

18. The method of claim 1, further comprising the step of dynamically modifying a specification of which details to collect in an iterative process.

19. The method of claim 1, further comprising the step of connecting to a running version of said software system.

20. The method of claim 1, further comprising the step of visually analyzing said collected details.

21. The method of claim 1, further comprising the step of visually analyzing said collected details for a plurality of instances of said program task.

22. The method of claim 1, further comprising the step of quantitatively analyzing said collected details.

23. The method of claim 1, further comprising the step of quantitatively analyzing said collected details for a plurality of instances of said program task.

24. A method for tracing details associated with a program task executing in a software system, comprising:

monitoring said software system to identify said program task; and

tracing details associated with said program task.

25. The method of claim 24, wherein a duration of said program task is defined by one or more conditions associated with a state of said software system.

26. The method of claim 25, wherein said one or more conditions is selected from the group consisting essentially of (i) an entry or exit of at least one specified method, (ii) a creation or deletion of at least one specified object, (iii) an invocation of at least one specified object, (iv) a passing of at least one specified object or scalar value as an argument, return value or field value, (v) at least one specified sequence of method invocations, and (vi) at least one specified resource exceeding at least one specified threshold.

27. The method of claim 24, wherein said collected details include at least one of the following: (i) an existence or sequence of specified method invocations, (ii) an existence or sequence of specified object creations and deletions, (iii) an existence or sequence of specified class loading and unloading, (iv) values of specified arguments to invocations of specified methods; (v) values of specified return values from invocations of specified methods, and (v) values of specified field values for invoked objects or field values for passed arguments.

28. The method of claim 24, further comprising the step of collecting said details for at least one of at least one specified number of task instances and at least one specified number of threads.

29. The method of claim 24, further comprising the step of dynamically modifying said program task specification associated with said analysis in an iterative process.

30. The method of claim 24, further comprising the step of dynamically modifying a specification of which details to collect in an iterative process.

31. The method of claim 24, further comprising the step of connecting to a running version of said software system.

32. A system for analyzing behavior of a software system, comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

collect details associated with a program task associated with said software system; and

provide said collected details for analysis.

33. A system for tracing details associated with a program task executing in a software system, comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

monitor said software system to identify said program task; and

trace details associated with said program task.

34. An article of manufacture for analyzing behavior of a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to collect details associated with a program task associated with said software system; and

a step to provide said collected details for analysis.

35. An article of manufacture for tracing details associated with a program task executing in a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to monitor said software system to identify said program task; and

a step to trace details associated with said program task.